Image Captioning via Masked Conditional Diffusion

Jiayi Zhou
M225783

Supervisors: Prof. Yasuhiko Morimoto
Prof. Sayaka Kamei
Prof. Koji Eguchi

Graduate School of Advanced Science and Engineering
Hiroshima University

This dissertation is submitted for the degree of
Master of Informatics and Data Science

Social Computing Laboratory August 2024

Acknowledgements

I would like to thank my supervisor, Professor Yasuhiko Morimoto, for his kind guidance, and constructive advice throughout my two-year Master's journey. His guidance was crucial in every aspect of my academic career, including finding research interests and delivering clear presentations.

Also, I appreciate the encouragement and useful suggestions from Associate Professor Sayaka Kamei and Professor Koji Eguchi.

Finally, My life at Hiroshima University has been greatly enriched by the past and present members of the social computing lab in the C702 room including Dr. Huidong Tang, Huachong Yu, Shuai Jiang, Jiale Lv, and Yuruo Su. I am grateful to all laboratory members and all faculty of the Graduate School of Advanced Science and Engineering in Hiroshima University.

Abstract

Current image captioning methods mainly adopt the autoregressive approach, where the sentences are predicted word-by-word. Despite the success of diffusion models (a non-autoregressive model) in image generation, their potential in image captioning remains underexplored due to the discrete nature of language and the continuity and redundancy of images. In this work, we present a masked conditional diffusion model (MC-Diffusion). It is based on a discrete denoising diffusion probabilistic model (D3PM) parameterized through a Markov Chain, conditioned on a vector quantized variational autoencoder (VQ-VAE) to extract discrete image features. Specifically, we first extract holistic discrete features via VQ-VAE for each image. With this comprehensive understanding of discrete features serving as conditions, the discrete diffusion model could be trained through cascaded transformer blocks and generate sentences in a non-autoregressive manner. Furthermore, we propose a masked condition strategy as a better substitute for classifier-free guidance. In classifier-free guidance, part of the data is trained without condition while others are trained with condition. Our masked condition technique, however, masks some partitions within the condition. Experiments on the CUB-200 dataset show that the proposed model performs better than the autoregressive baseline on several metrics. Compared to classifier-free guidance, our masked condition strategy achieves similar performance on CLIPscore while alleviating the hurt on reference-based metrics (e.g. BLEU, Meteor, etc.).

Table of contents

List of figures

List of tables

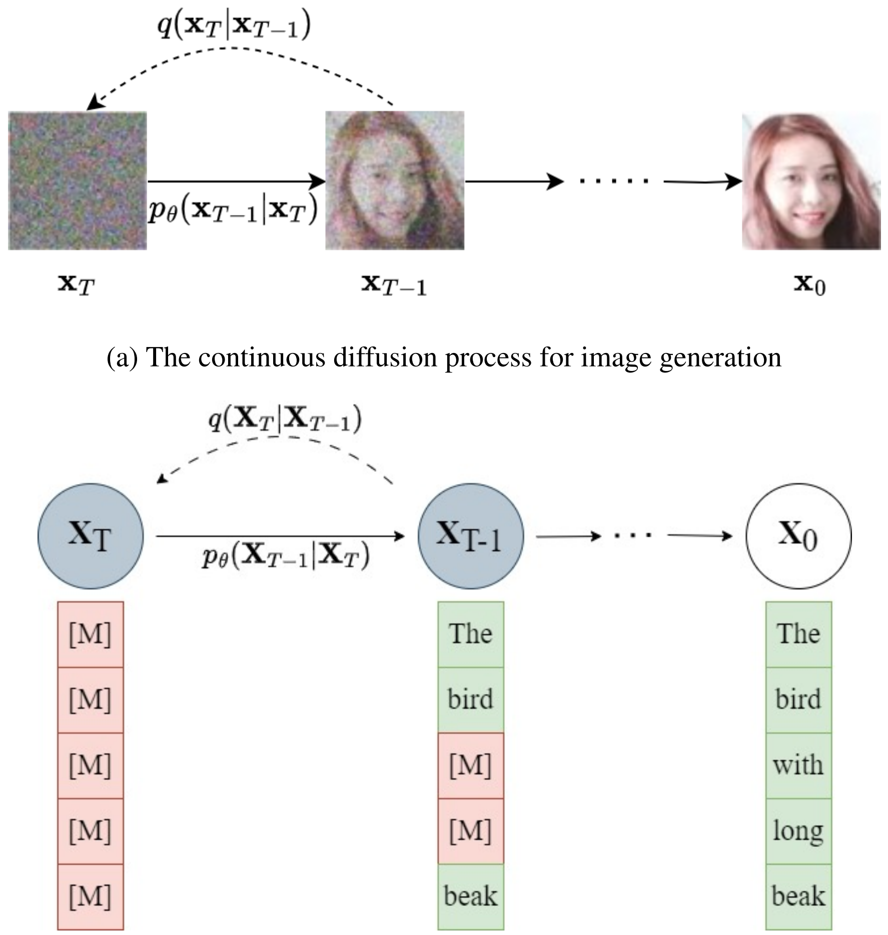
Chapter 1

Introduction

Image captioning aims to generate natural language descriptions of an image. This multimodal task bridges the realms of natural language processing and computer vision, requiring the model to meticulously capture both global and local features of an image. Autoregressive model[6] stands out as a widely used approach for image captioning. Based on the autoregressive model, current state-of-the-art models [13, 9, 45, 26, 46] have achieved remarkable results in image captioning by capitalizing on the encoder-decoder structure. Specifically, an image encoder transforms the image into high-level semantics, and a text decoder is utilized to generate sentences word-by-word sequentially. However, such techniques suffer from unidirectional semantic passing issues. If a wrong word is predicted, incorporating this mistake as input will propagate the error to the subsequent sentence.

To mitigate the weakness above, recent advances of non-autoregressive image captioning approaches[50, 30] which allow for bidirectional semantic passing are motivated by the success of diffusion model in image generation. Diffusion models[21, 38] are a generative model that has laid the milestone for image generation, producing varied results while maintaining visual fidelity. With the pioneering work of Bit diffusion[7], which explores continuous diffusion model to generate discrete texts with each word represented as binary bits, SCD-Net[30] is able to generate image descriptions with the relevant semantic conditions.

Despite their success, existing diffusion-based image captioning methods still have limitations that require improvement. One issue is the image representation. Images are data with high-level redundancy, performing low information density; Text is discrete with highly abstract information, it can describe images concisely. Existing methods extract the continuous image representation as an input of the discrete text decoder. However, it is non-trivial to align continuous image representation and discrete text feature well. Discrete representations are a more natural fit for many modalities we are interested in[40]. Hence, for better feature alignment, discrete representations may be better for image captioning.


$q(\mathbf{x}_T|\mathbf{x}_{T-1})$
$p_\theta(\mathbf{x}_{T-1}|\mathbf{x}_T)$
$\mathbf{x}_T$
$\mathbf{x}_{T-1}$
$\mathbf{x}_0$
$q(\mathbf{X}_T|\mathbf{X}_{T-1})$
$p_\theta(\mathbf{X}_{T-1}|\mathbf{X}_T)$
XT
XT-1
X0
[M]
[M]
[M]
[M]
[M]
The
bird
[M]
[M]
beak
The
bird
with
long
beak

(a) The continuous diffusion process for image generation

(b) The discrete diffusion process for text generation. The noise depends on the transition matrix. [M] denotes mask token.

Fig. 1.1 Two types of diffusion models

Another issue is the classifier-free guidance[22]. Classifier-free guidance improves $p(\text{image}|\text{caption})$ at the expense of $p(\text{caption}|\text{image})$ in image captioning, which demonstrates the trade-off between image description capability and grammatical accuracy[24]. DDCap[50] exploits classifier-free guidance by removing image conditions from some training data. Nevertheless, classifier-free guidance may lead to a great decrease in reference-based metrics which measure the similarity between generated and human-written captions (e.g. BLEU, ROUGE, etc.). The performance degradation can be attributed to simply removing the image condition in some examples may not allow the model to learn a comprehensive understanding beyond low-level features, i.e. worse alignment between features.

We devise a Masked Conditional diffusion model (MC-Diffusion) to address the limitations. First, we leverage the Vector Quantised Variational AutoEncoder (VQ-VAE)[40] as an image encoder to convert an image into discrete image tokens. These tokens are then used as conditions to guide the discrete diffusion model in generating descriptions. Roughly speaking, as shown in figure1.1(b), the forward process of the discrete diffusion model is parameterized by a Markov chain, corrupting the original sentence into noise through a transition matrix. Starting from noisy data, the inference process gradually denoising towards the desired caption.

Secondly, thanks to the discrete image tokens modeled by VQ-VAE, we introduce a masked condition strategy: for each data sample, a portion of the discrete image tokens are removed before input into the discrete diffusion model as the condition. In contrast to MAE[19], where masking occurs on small patches of an image before being processed by a vision transformer encoder, our work leverages the advantage of VQ-VAE in modeling discrete latent space, enabling masking on crucial image tokens spanning across multiple pixels. Consequently, this poses a more challenging task for MC-Diffusion, as it needs a more comprehensive understanding beyond low-level image features and better alignment. In addition, the masked condition strategy can be regarded as another form of classifier-free guidance with a lower expense of $p(\text{caption}|\text{image})$, as it involves removing a portion of discrete image tokens rather than discarding the entire image data for certain examples.

To assess the performance of the MC-Diffusion, we conduct image captioning experiments with the CUB-200[44] dataset. Our method achieves better results on all reference-based metrics than the autoregressive baseline. We further verify the proposed masked condition strategy, experiment results confirm our idea that our masked condition strategy achieves similar performance on CLIPscore[20] while alleviating the hurt of reference-based metrics.

To sum up, the main contributions of our work are as follows.

- MC-Diffusion designs a novel model structure for image captioning. There is a VQ-VAE encoder to extract discrete image tokens and a discrete diffusion model to generate caption.
- MC-diffusion also paves a new way to guide the discrete diffusion model with masked condition strategy.

Chapter 2

Related Work

2.1 Autoregressive Models

The early image captioning methods[8, 11, 32] focus on utilizing a convolutional neural network (CNN) as an encoder to learn high-level image representations followed by a recurrent neural network (RNN) to predict the sentence word-by-word. Later on, techniques[2, 23] leverage attention mechanism are explored to predict the caption by concentrating on relevant image region. Sparked by the advantage of bilinear pooling[15] in capturing more discriminative representation, X-LAN[33] is able to capture 2^{nd} interaction between multimodal features.

After that, with the triumph of Transformer[41] in the NLP field, M^2 transformer[9], which is one of the current state-of-the-art methods, exploit mesh-like connectivity to extract both low and high level features. RSTNet[49] enhances the transformer decoder with the adaptive-attention module to measure the importance of visual-language prior. DLCT[31] explores the intrinsic properties of descriptive region features and traditional grid features by an advanced Dual-way self-attention layer. CaMEL[5] is an approach with two different but interconnected Transformer models, performing mean teacher learning with knowledge distillation to learn from each other. The text decoder of the above methods generates text in a left-to-right reading order, where the prediction of the next word is always based on the words generated in previous steps.

2.2 Non-autoregressive Models

Unlike the autoregressive methods mentioned, non-autoregressive methods predict each word independently of previously generated ones. A look back mechanism[14] is introduced for variable refinement and faster caption generation. NAIC[18] first trains a non-autoregressive model in a reinforcement learning mode to maximize a sentence-level reward. Yu et al.[48] develop an end-to-end model by adopting the Swin-Transformer with a semantic retrieval module to increase the decoder input scale, which achieved state-of-the-art performance. SAIC[47] makes a trade-off between captioning speed and model performance by a semi-autoregressive model which jumpily predicts some words by autoregressive module while other skipped words by non-autoregressive step. MNIC[16] proposes masked non-autoregressive techniques to generate more diverse captions, it is worth noting that the idea of MNIC coincides with the subsequent application of the D3PM[3] with an absorbing state.

Most recently, a non-autoregressive method called diffusion models first proposed in [37] made a huge success in image generation[10, 36, 36, 17]. As shown in Fig1.1, depending on the data type, diffusion models can be divided into continuous diffusion models[21, 38] and discrete diffusion models[3, 50]. On one hand, inspired by the continuous diffusion models, Bit Diffusion[7] projects tokens into continuous space and generates text using a continuous diffusion model. SCD-Net[30] further improves Bit Diffusion by better visual-language alignment through cascaded transformer blocks. On the other hand, DDCap[50] first utilizes a discrete diffusion model with a Contrastive Language-Image Pretraining[35] (CLIP) image encoder in the image captioning task.

Our work also falls into the discrete diffusion model for image captioning. A new masked condition strategy is designed to guide the diffusion model with better feature alignment.

Chapter 3

Preliminaries

3.1 Loss Functions of Continuous Diffusion Models

Diffusion models focus on modeling the data distribution $p(\mathbf{x}_0)$ through maximum likelihood estimation: maximize the likelihood $p(\mathbf{x}_0)$ of all observed $\mathbf{x}_0$:

$$\text{Objective: } \arg\max_{\theta} \log p(\mathbf{x}_0) \tag{3.1}$$

where θ denotes the model parameter. Recall **E**vidence **L**ower **Bo**und (ELBO):

$$\begin{aligned}
\log p(\mathbf{x}_0) &= \log \int p(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \\
&= \log \int \frac{p(\mathbf{x}_{0:T}) q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T} \\
&= \log \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}] \\
&\geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}]
\end{aligned} \tag{3.2}$$

Note the last step is based on the Jenson Inequality. $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ denotes the joint distribution of forward process from time step $t = 1$ to $t = T$ conditioned on raw data $\mathbf{x}_0$, $p(\mathbf{x}_{0:T})$ denotes the joint distribution of data from time step $t = 0$ to $t = T$, which could be parameterized as:

$$p(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{3.3}$$

where $p(\mathbf{x}_T)$ is sampled from noise distribution $N(\mathbf{0}, \mathbf{I})$, $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ refers to the denoising neural network to predict $\mathbf{x}_{t-1}$ from $\mathbf{x}_t$ for one step. Furthermore, we can maximize the lower bound Eq3.2 rather than the direct objective 3.1:

$$\begin{aligned}
&\arg\max_{\theta} \log p(\mathbf{x}) \\
\propto&\arg\max_{\theta} \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}[\log \frac{p(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}] \\
=&\arg\max_{\theta} \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]}_{L_0} - \underbrace{D_{KL}[q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)]}_{L_T} \\
&\quad - \underbrace{\sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]}_{L_{t-1}} \qquad (3.4) \\
\propto&\arg\min_{\theta} \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))] \qquad (3.5)
\end{aligned}$$

From Eq3.4 to Eq3.5, the term L_T can be seen as a constant because it has no learnable parameters and term L_0 can also be ignored for two reasons: 1. it can be approximated by the same network in L_{t-1}. 2. it makes better performance and simple to implement. Furthermore, with the summation over time step t is equivalently approximated by Monte Carlo estimation:

$$\arg\min_{\theta} \mathbb{E}_{t\sim U(2,T)} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))] \qquad (3.6)$$

By further derivation, we have the following three equivalent loss functions:

$$L_{\text{simple}} = \mathbb{E}_{t,\mathbf{x}_0}||\hat{\mathbf{x}}_\theta(\mathbf{x}_t,t) - \mathbf{x}_0||_2^2 \qquad (3.7)$$

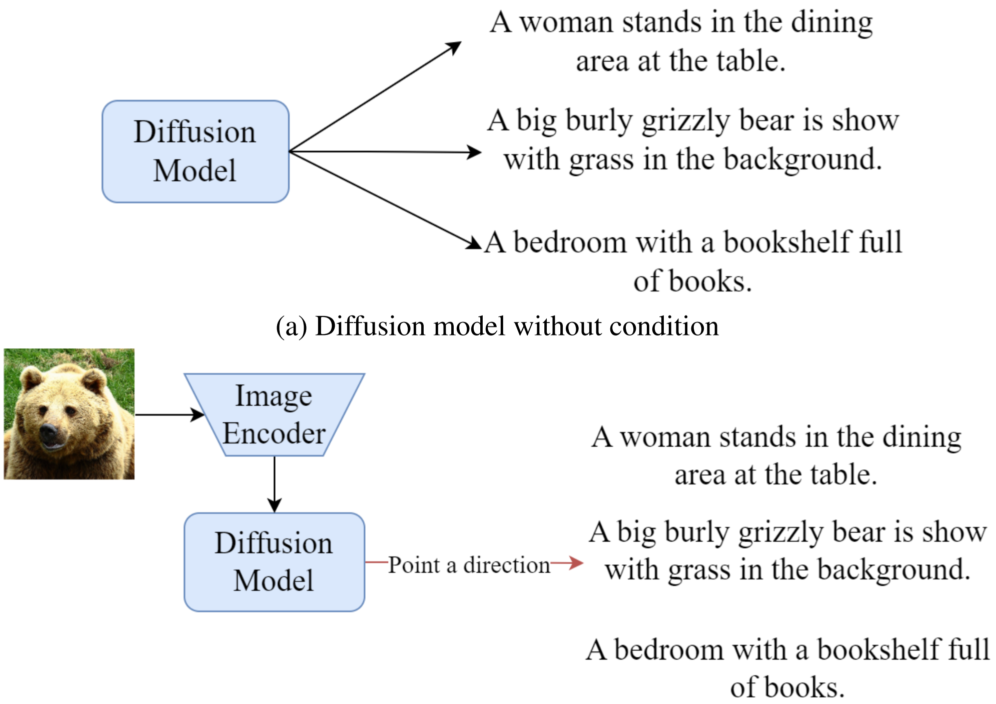
$$L_{\text{simple}} = \mathbb{E}_{t,\mathbf{x}_0}||\hat{\varepsilon}_\theta(\mathbf{x}_t,t) - \varepsilon_0||_2^2 \qquad (3.8)$$

$$L_{\text{simple}} = \mathbb{E}_{t,\mathbf{x}_0}||\hat{\mathbf{s}}_\theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)||_2^2 \qquad (3.9)$$

Eq3.7 refers to estimate the raw data $\mathbf{x}_0$ with neural network $\hat{\mathbf{x}}_\theta(\mathbf{x}_t,t)$ which takes the noise data $\mathbf{x}_t$ and time embedding t as input. Similarly, Eq3.8 and Eq3.9 can be interpreted as using neural network $\hat{\varepsilon}_\theta(\mathbf{x}_t,t)$ and $\hat{\mathbf{s}}_\theta(\mathbf{x}_t,t)$ to approximate the noise ε_0 added to the raw data and the gradient of the log of noise data $\mathbf{x}_t$.

3.2 Guidance in Diffusion Models

Diffusion models enable the sampling of random data points from the learned distribution $p(\mathbf{x})$. However, in image captioning, as shown in Fig3.1, our goal is often to control the generated data explicitly rather than generating random data that may not meet our interests.


Diffusion Model
A woman stands in the dining area at the table.
A big burly grizzly bear is show with grass in the background.
A bedroom with a bookshelf full of books.
Image Encoder
Diffusion Model
Point a direction
A woman stands in the dining area at the table.
A big burly grizzly bear is show with grass in the background.
A bedroom with a bookshelf full of books.

(a) Diffusion model without condition

(b) Diffusion model with condition

Fig. 3.1 Generate interested data in Diffusion models

A natural idea is to learn the conditional distribution $p(\mathbf{x}|\mathbf{y})$ shown in Fig3.1(b), i.e., to take the image through the image encoder and extract features that are used as additional inputs in the text decoder of the diffusion model. However, experiments demonstrate that such a paradigm will generate data less relevant to the condition information[29, 39]. Later on, researchers developed Classifier guidance[10] and Classifier-free guidance[22] to better control the synthesized data.

3.2.1 Classifier Guidance

Here, we start by letting the model estimate the gradient of the log conditional distribution $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$ in Eq3.9,

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) &= \nabla_{\mathbf{x}_t} \log \frac{p(\mathbf{x}_t,\mathbf{y})}{p(\mathbf{y})} \\
&= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t,\mathbf{y}) - \nabla \mathbf{x}_t \log p(\mathbf{y}) \\
&= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) p(\mathbf{y}|\mathbf{x}_t) \\
&= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)
\end{aligned} \tag{3.10}$$

Note that the equation from the second to the third line is because $\log p(\mathbf{y})$ is independent of $\mathbf{x}_t$, so the gradient term equals to 0. Eq3.10 reveals that training a conditional model $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$ is inherently training a unconditional model $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ plus the gradient of a classifier $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$. To manipulate how relevant the generated data is to the conditional information, the authors rearrange the Eq3.10 by introducing a hyperparameter γ as:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) := \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \tag{3.11}$$

Nevertheless, the classifier guidance requires us to train an additional classifier, meaning that it consumes additional computational resources, so researchers have devised classifier-free guidance to guide the diffusion models without a classifier.

3.2.2 Classifier-free Guidance

Let's begin with the classifier guidance formula Eq3.11,

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) : &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \\
&= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \gamma(\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)) && (3.12) \\
&= (1-\gamma)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) && (3.13)
\end{aligned}$$

As shown in Eq3.13, classifier-free guidance learns a conditional diffusion model $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$ and a unconditional diffusion model $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$. A higher γ value indicates that the model takes more account of conditional information, with $\gamma = 1$ representing a fully conditional diffusion model. In the training step, the conditional model and unconditional model are trained in one model by randomly replacing the condition with zeros.

Another interpretation goes to Eq3.12 where the term $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ represents a directional vector in high-dimensional data space, pointing from the unconditional distribution $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ to conditional distribution $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$. The hyperparameter γ adjusts the scale of this direction. Classifier guidance at this point biases the original unconditional data distribution to the new data distribution according to this directional vector.

Kornblith et al.[24] rearrange Eq3.12 in image captioning as,

$$\begin{aligned}\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{y}) &= \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)+\gamma(\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{y})-\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t))\\ &= \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)+\gamma\nabla_{\mathbf{x}_t}\log\frac{p(\mathbf{x}_t|\mathbf{y})}{p(\mathbf{x}_t)}\\ &= \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)+\gamma\nabla_{\mathbf{x}_t}\log\frac{p(\mathbf{y}|\mathbf{x}_t)}{p(\mathbf{y})}\\ &= \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)+\gamma\nabla_{\mathbf{x}_t}\log p(\mathbf{y}|\mathbf{x}_t) \qquad (3.14)\end{aligned}$$

They interpret Eq3.14 as follows: when $\gamma = 1$, the model learns a complete conditional model , but when $\gamma > 1$, it enlarge the $p(\mathbf{y}|\mathbf{x}_t)$. This represents a trade-off between $p(\text{image}|\text{caption})$ and $p(\text{caption}|\text{image})$. Notice that Eq3.11 and Eq3.14 have the same form, but they have different meanings: Eq3.11 is meant to guide the diffusion model with a classifier while Eq3.14 emphasizes the implicit image generation model. They differ in the meaning of conditional information $\mathbf{y}$, one is class information and the other is the image itself.

3.3 Vector quantized variational autoencoder

The Vector Quantized Variational Autoencoder (VQVAE)[40] learns to extract discrete representations from high-dimensional data (e,g. image and video) and then reconstruct them. As shown in figure4.1, the encoder $\mathbf{e} = E(\mathbf{x}) \in \mathbb{R}^{h\times w\times d}$ first compresses the given image sample $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ into latent features, followed by a discretization operator $Q(\mathbf{e},\mathbf{Y})$ to map each feature with the closest embedding in codebook $\mathbf{Y} = \{y_k\}_{k=1}^{K}$:

$$y_{ij} = \underset{y_k\in\mathbf{Y}}{argmin}\|e_{ij}-y_k\|_2^2 \qquad (3.15)$$

where $e_{ij} \in \mathbb{R}^d$ and $y_{ij} \in \mathbb{R}^d$ represents (i, j) entry of the encoded image feature $\mathbf{e} \in \mathbb{R}^{h\times w\times d}$ and the output of discretized image tokens $\mathbf{y} \in \mathbb{R}^{h\times w\times d}$, respectively. H and W denote the image height and width, whereas h and w denote the height and width of the encoded image feature. Hence, the decoder D can reconstruct the data by $\bar{\mathbf{x}} = D(\mathbf{y})$. VQ-VAE is trained via the following objective:

$$L = \|\mathbf{x}-D(\mathbf{y})\|_2^2+\|\mathrm{sg}[E(\mathbf{x})]-\mathbf{y}\|_2^2+\beta\|\mathrm{sg}[\mathbf{y}]-E(\mathbf{x})\|_2^2 \qquad (3.16)$$

where sg refers to the stop-gradient operator. The second term of Equation 3.16 is the codebook loss, which minimizes the distance between the output of the encoder and codebook embedding.

Roughly speaking, embeddings in the codebook represent various discrete features, such as species, age, and gender in the image. Replacing the output features of the encoder with the embeddings from the codebook is equivalent to removing redundant information from the image, thereby increasing the information density of the output features.

Chapter 4

Masked Conditional Diffusion Model

4.1 Model structure

As shown in figure 4.1, the MC-Diffusion is constructed as a VQ-VAE image encoder and a discrete diffusion text decoder. Given an image, the VQ-VAE encoder first encodes the image into latent representations. Then it maps this representation with the closest embedding in the codebook as condition $\mathbf{y}$. The discrete diffusion text decoder contains several transformer blocks with a self-attention, a cross-attention, a feed-forward layer, and an Adaptive Layer Normalization layer. The decoder integrates the condition $\mathbf{y}$ with the current text representation $\mathbf{x}_t$ in the cross-attention layer and predicts the noiseless text sequence $\mathbf{x}_0$. Furthermore, the masked condition strategy is applied between the image encoder and discrete diffusion decoder to randomly mask some discrete image tokens in the condition.

4.2 Discrete Diffusion Model

Given an image-caption pair, MC-Diffusion begins with a VQ-VAE to extract discrete image tokens, represented as $\mathbf{y} \in \mathbb{R}^{h \times w}$ and reshaped into a vector $\mathbf{y} = \{y_1, y_2, ..., y_N\}$, where $N = hw$. Assuming the ground truth caption is $\mathbf{x}$, the MC-Diffusion aims to maximize $p(\mathbf{x}|\mathbf{y})$.

4.2.1 Forward Process

Unlike continuous diffusion models that corrupt an image by gradually injecting Gaussian noise, discrete diffusion models corrupt text by randomly replacing some of the tokens with other tokens or [MASK] token step-by-step through a Markov chain.

Specifically, consider the one-hot version of a token $\mathbf{x}_{t-1} \in \mathbb{R}^{1 \times (M+1)}$, where t denotes the time step, and M+1 denotes the size of vocabulary plus the [MASK] token. The forward

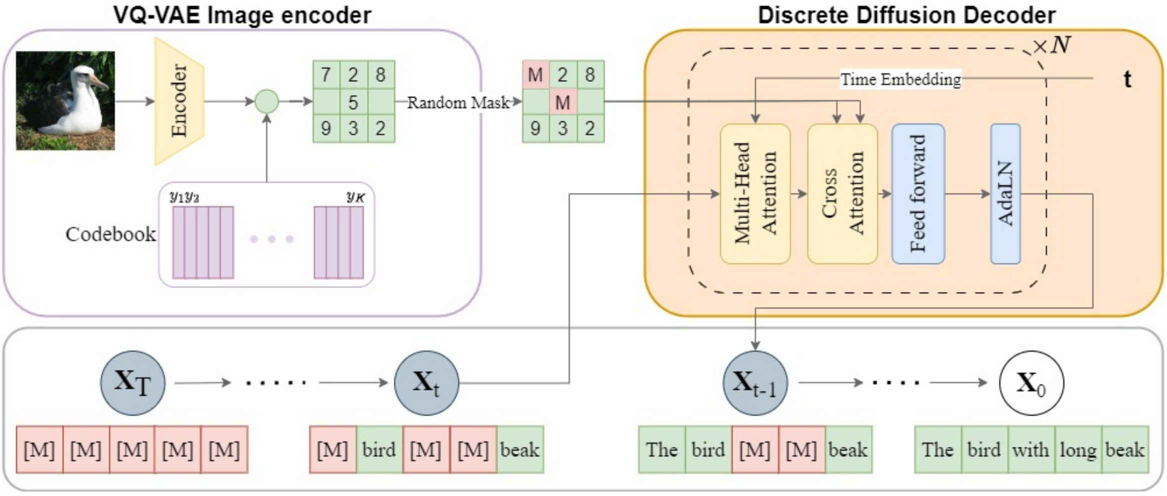


Fig. 4.1 Overall framework of Masked Conditional Diffusion Model (MC-Diffusion).

process can be parameterized by a transition matrix $\mathbf{Q}_t \in \mathbb{R}^{(M+1)\times(M+1)}$ and $[\mathbf{Q}_t]_{ij} = q(\mathbf{x}_t = j|\mathbf{x}_{t-1} = i)$. To reduce ambiguity, $\mathbf{X}_t$ denotes the random variable and $\mathbf{X}_t = \mathbf{x}_t$ its realisation. The one-step transition can be written as

$$q(\mathbf{X}_t|\mathbf{X}_{t-1} = \mathbf{x}_{t-1}) = \mathbf{x}_t\mathbf{Q}_t \in \mathbb{R}^{1\times(M+1)} \tag{4.1}$$

where $q(\mathbf{X}_t|\mathbf{X}_{t-1} = \mathbf{x}_{t-1})$ is the probability distribution over the different tokens $\mathbf{X}_t$ given the previous time token $\mathbf{x}_{t-1}$. $\mathbf{x}_t\mathbf{Q}_t$ refers to the row vector which is the product of row vector $\mathbf{x}_t \in \mathbb{R}^{1\times(M+1)}$ and matrix $\mathbf{Q}_t \in \mathbb{R}^{(M+1)\times(M+1)}$. Notably, $\mathbf{x}_{t-1}$ represents a single token; we assume that the transition equation Eq4.1 is applied to each token independently in a text.

Similar to the continuous diffusion process, t-step marginal can also be parameterized from $\mathbf{x}_0$:

$$q(\mathbf{X}_t|\mathbf{X}_0 = \mathbf{x}_0) = \mathbf{x}_0\bar{\mathbf{Q}}_t \in \mathbb{R}^{1\times(M+1)}, \quad \text{with} \quad \bar{\mathbf{Q}}_t = \mathbf{Q}_1\mathbf{Q}_2...\mathbf{Q}_t \tag{4.2}$$

Given the noiseless token $\mathbf{x}_0$, Eq4.2 enables us tp directly compute the noise token $\mathbf{x}_t$. There are multiple choices for the transition matrix $\mathbf{Q}_t$. Here, we follow Mask-and-replace diffusion strategy from VQ-Diffusion[17],

$$\mathbf{Q}_t = \begin{pmatrix} \alpha_t+\beta_t & \beta_t & \beta_t & \dots & 0 \\ \beta_t & \alpha_t+\beta_t & \beta_t & \dots & 0 \\ \beta_t & \beta_t & \alpha_t+\beta_t & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \dots \\ \gamma_t & \gamma_t & \gamma_t & \dots & 1 \end{pmatrix} \tag{4.3}$$

This transition matrix can be interpreted as follows: additional [MASK] token is introduced, each token has a probability of γ_t to be replaced by [MASK] token, β_t to transfer to another token in the vocabulary, and $\alpha_t = 1 - M\beta_t - \gamma_t$ to be unchanged.

It can be proved that this Markov process has a stationary distribution, which means that after unlimited steps, the original one-hot distribution will become:

$$p(x_T) = [\bar{\beta}_T, \bar{\beta}_T, \dots, \bar{\beta}_T, \bar{\gamma}_T]^\intercal \tag{4.4}$$

where $\bar{\gamma}_t = 1 - \prod_{i=1}^{t}(1-\gamma_i)$, $\bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i$ and $\bar{\beta}_t = (1 - \bar{\alpha}_t - \bar{\gamma}_t)/K$ can be computed and stored as noise schedule. In MC-Diffusion, $\bar{\gamma}_t$ is set to linearly increased from 0 to 0.9 and $\bar{\beta}_t$ from 0 to 0.1.

4.2.2 Reverse Process

Starting with the noise tokens $\mathbf{x}_T$ sampled from the probability distribution Eq4.4 and the image condition $\mathbf{y}$, we can write the one-step reverse process:

$$\begin{aligned}
q(\mathbf{X}_{t-1}|\mathbf{X}_t = \mathbf{x}_t, \mathbf{y}) &= \frac{q(\mathbf{X}_{t-1}, \mathbf{X}_t = \mathbf{x}_t|\mathbf{y})}{q(\mathbf{X}_t = \mathbf{x}_t|\mathbf{y})} \\
&= \frac{\sum_{\mathbf{x}_0} q(\mathbf{X}_{t-1}, \mathbf{X}_t = \mathbf{x}_t, \mathbf{X}_0 = \mathbf{x}_0|\mathbf{y})}{q(\mathbf{X}_t = \mathbf{x}_t|\mathbf{y})} \\
&= \frac{q(\mathbf{X}_t = \mathbf{x}_t|\mathbf{y}) \sum_{\mathbf{x}_0} q(\mathbf{X}_{t-1}, \mathbf{X}_0 = \mathbf{x}_0|\mathbf{X}_t = \mathbf{x}_t, \mathbf{y})}{q(\mathbf{X}_t = \mathbf{x}_t|\mathbf{y})} \\
&= \sum_{\mathbf{x}_0} q(\mathbf{X}_{t-1}|\mathbf{X}_t = \mathbf{x}_t, \mathbf{X}_0 = \mathbf{x}_0, \mathbf{y}) q(\mathbf{X}_0 = \mathbf{x}_0|\mathbf{X}_t = \mathbf{x}_t, \mathbf{y})
\end{aligned} \tag{4.5}$$

Note that we do not have $q(\mathbf{X}_0 = \mathbf{x}_0|\mathbf{x}_t = \mathbf{x}_t, \mathbf{y})$ but can be approximated by neural network $p_\theta(\mathbf{X}_0 = \hat{\mathbf{x}}_0|\mathbf{X}_t = \mathbf{x}_t, \mathbf{y})$. In MC-Diffusion, $p_\theta(\mathbf{X}_0 = \hat{\mathbf{x}}_0|\mathbf{X}_t = \mathbf{x}_t, \mathbf{y})$ is several transformer blocks with input $\mathbf{x}_t$ and $\mathbf{y}$:

$$\begin{aligned}
q(\mathbf{X}_{t-1}|\mathbf{X}_t = \mathbf{x}_t, \mathbf{y}) &\approx p_\theta(\mathbf{X}_{t-1}|\mathbf{X}_t = \mathbf{x}_t, \mathbf{y}) \\
&= \sum_{\mathbf{x}_0} q(\mathbf{X}_{t-1}|\mathbf{X}_t = \mathbf{x}_t, \mathbf{X}_0 = \mathbf{x}_0, \mathbf{y}) p_\theta(\mathbf{X}_0 = \mathbf{x}_0|\mathbf{X}_t = \mathbf{x}_t, \mathbf{y})
\end{aligned} \tag{4.6}$$

where the posterior is

$$q(\mathbf{X}_{t-1}|\mathbf{X}_t=\mathbf{x}_t,\mathbf{X}_0=\mathbf{x}_0)=\frac{q(\mathbf{X}_t=\mathbf{x}_t|\mathbf{X}_{t-1},\mathbf{X}_0=\mathbf{x}_0)q(\mathbf{X}_{t-1}|\mathbf{X}_0=\mathbf{x}_0)}{q(\mathbf{X}_t=\mathbf{x}_t|\mathbf{X}_0=\mathbf{x}_0)} \tag{4.7}$$

$$=\frac{q(\mathbf{X}_t=\mathbf{x}_t|\mathbf{X}_{t-1})q(\mathbf{X}_{t-1}|\mathbf{X}_0=\mathbf{x}_0)}{q(\mathbf{X}_t=\mathbf{x}_t|\mathbf{X}_0=\mathbf{x}_0)} \tag{4.8}$$

$$=\frac{\mathbf{x}_t\mathbf{Q}_t^\intercal\odot\mathbf{x}_0\bar{\mathbf{Q}}_{t-1}}{\mathbf{x}_0\bar{\mathbf{Q}}_t\mathbf{x}_t^\intercal} \tag{4.9}$$

where $\odot$ denotes element-wise multiplication. We use the property of the Markov Chain when deriving from Eq4.7 to Eq4.8.

In sum, the reverse process of the discrete diffusion model first samples a random token sequence $\mathbf{x}_T$ from distribution Eq4.4 and encodes an image into discrete image tokens $\mathbf{y}$. Then input $\mathbf{x}_T$ and $\mathbf{y}$ into the transformer blocks $p_\theta(\mathbf{X}_0=\hat{\mathbf{x}}_0|\mathbf{X}_t=\mathbf{x}_t,\mathbf{y})$ to predict the noiseless token sequence $\hat{\mathbf{x}}_0$. Finally, the $\mathbf{x}_T$, $\mathbf{y}$ and predicted $\hat{\mathbf{x}}_0$ are used to compute $\mathbf{x}_{T-1}$ according to Eq4.6 and the process above is repeated until we have the $\mathbf{x}_0$.

4.2.3 Loss Function

Different from DDPM[21] that directly predict the added noise L_{simple}[3.8], we follow D3PM[3] to minimize the L_{vlb}[3.6] plus an auxiliary loss which encourages to predict better $\mathbf{x}_0$:

$$L=\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]+\lambda\mathbb{E}_{q(\mathbf{x}_0)}\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\log p_\theta(\mathbf{x}_0|\mathbf{x}_t,\mathbf{y}) \tag{4.10}$$

4.3 Masked Condition Strategy

Classifier-free guidance[22] has shown the trade-off between variety and quality in image generation. In image captioning, as discussed in section3.2.2, it also makes a trade-off between $p(\text{caption}|\text{image})$ and $p(\text{image}|\text{caption})$. Recall classifier-free guidance Eq3.13:

$$\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{y}):=(1-\gamma)\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)+\gamma\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{y}) \tag{4.11}$$

It trains a conditional model $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{y})$ and an unconditional model $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)$ at the same time by removing the condition information in some training samples. Our masked condition strategy, however, does not remove all condition information but masks some entries in the unconditional model $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)$. To do this, each discrete token $y_i\in\mathbf{y}=$

$\{y_1, y_2, ..., y_N\}$ has a probability P_{mask} to be masked:

$$y_i = \begin{cases} [\text{MASK}], p \leq P_{\text{mask}} \\ y_i, p > P_{\text{mask}} \end{cases} \tag{4.12}$$

where p is a random number sampled from $U(0,1)$. Then the condition is divided into $\mathbf{y}_{[\text{M}]}$ representing the [MASK] tokens and $\mathbf{y}_{\text{other}}$ denoting the tokens not masked.

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) := \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_{[\text{M}]}, \mathbf{y}_{\text{other}}) + (1-\gamma)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) \tag{4.13}$$

Eq4.13 is our masked condition strategy. We have a fully conditional model $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$ and a conditional model with mask $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_{[\text{M}]}, \mathbf{y}_{\text{other}})$. Once again like classifier-free guidance, γ controls how much we care about the condition information. As Algorithm 1 shows, in the training period, the two models $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$ and $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_{[\text{M}]}, \mathbf{y}_{\text{other}})$ are trained in one model by masking the condition in p_{CM} of the data in the dataset while leaving others fully conditioned.

By masking some condition information, the model is trained to align different image features with text at each time step. Hence, this strategy poses a more challenging problem to the neural network to learn the holistic understanding beyond low-level features.

Algorithm 1 Training of the MC-Diffusion with masked condition strategy

```
Require: Data with partial condition rate $P_{\text{part\_cond}} \in [0,1]$
  repeat
    (image, caption)←sampled from the training dataset
    $t \sim U(\{1,...,T\})$ ←sample a time step t
    $\mathbf{x}_0$ ←CLIP text embedding(caption), $\mathbf{y}$ ← VQ-VAE image Encoder(image)
    $\mathbf{x}_t$ ←add noise to $\mathbf{x}_0$                                              ▷ Eq4.2
    random number p ←sampled from $U(0,1)$
    if $p < P_{\text{part\_cond}}$ then
      $(\mathbf{y}_{[\text{M}]}, \mathbf{y}_{\text{other}})$ ←mask some entries in condition $\mathbf{y}$                ▷ Eq4.12
      $(\hat{\mathbf{x}}_0, \mathbf{x}_{t-1})$ ←discrete diffusion decoder $(\mathbf{y}_{[\text{M}]}, \mathbf{y}_{\text{other}}, \mathbf{x}_t)$   ▷ Eq4.6 and 4.9
      Loss← $(\hat{\mathbf{x}}_0, \mathbf{x}_{t-1}, \mathbf{x}_0)$                                  ▷ Eq4.10
    else
      $(\hat{\mathbf{x}}_0, \mathbf{x}_{t-1})$ ←discrete diffusion decoder $(\mathbf{y}, \mathbf{x}_t)$
      Loss← $(\hat{\mathbf{x}}_0, \mathbf{x}_{t-1}, \mathbf{x}_0)$                                  ▷ Eq4.10
    end if
  until converged
```

Algorithm 2 Inference of the MC-Diffusion with masked condition strategy

```
Require: Guidance scale $\gamma$, maximum time step T
  $\mathbf{x}_t \leftarrow$sampled from $p(\mathbf{x}_T)$                                   ▷Eq4.4
  $\mathbf{y} \leftarrow$ VQ-VAE image Encoder(image)
  for $t = T$ to 1 do
    $(\mathbf{y}_{[\mathrm{M}]}, \mathbf{y}_{\mathrm{other}}) \leftarrow$mask some entries in condition $\mathbf{y}$                ▷ Eq4.12
    $\mathbf{x}_{t-1,\mathrm{cond}} \leftarrow$discrete diffusion decoder $(\mathbf{y}, \mathbf{x}_t)$
    $\mathbf{x}_{t-1,\mathrm{part\_cond}} \leftarrow$discrete diffusion decoder $(\mathbf{y}_{[\mathrm{M}]}, \mathbf{y}_{\mathrm{other}}, \mathbf{x}_t)$
    $\mathbf{x}_{t-1} \leftarrow \gamma \mathbf{x}_{t-1,\mathrm{part\_cond}} + (1-\gamma)\mathbf{x}_{t-1,\mathrm{cond}}$                ▷ Eq4.13
  end for
  return Generated caption $\mathbf{x}_0$
```

If we further make some derivation on our masked condition strategy:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) &= \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}_{[\mathrm{M}]}, \mathbf{y}_{\mathrm{other}}) + (1-\gamma)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) \\
&= \gamma \nabla_{\mathbf{x}_t} \log \frac{p(\mathbf{x}_t, \mathbf{y}_{[\mathrm{M}]}, \mathbf{y}_{\mathrm{other}})}{p(\mathbf{y}_{[\mathrm{M}]}, \mathbf{y}_{\mathrm{other}})} + (1-\gamma)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) \\
&= \gamma[\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t, \mathbf{y}_{[\mathrm{M}]}, \mathbf{y}_{\mathrm{other}}) - \nabla_{\mathbf{x}_t} \log p(\mathbf{y}_{[\mathrm{M}]}, \mathbf{y}_{\mathrm{other}})] + (1-\gamma)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) \\
&= \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t, \mathbf{y}_{[\mathrm{M}]}, \mathbf{y}_{\mathrm{other}}) + (1-\gamma)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) \\
&= \gamma \nabla_{\mathbf{x}_t} \log p(\mathbf{y}_{[\mathrm{M}]}|\mathbf{x}_t, \mathbf{y}_{\mathrm{other}}) p(\mathbf{x}_t, \mathbf{y}_{\mathrm{other}}) + (1-\gamma)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) \\
&= \gamma[\underbrace{\nabla_{\mathbf{x}_t} p(\mathbf{y}_{[\mathrm{M}]}|\mathbf{x}_t, \mathbf{y}_{\mathrm{other}})}_{\text{Image Inpainting Model}} + \underbrace{\nabla_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{y}_{\mathrm{other}})}_{\text{Partial Caption Model}}] + (1-\gamma)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})
\end{aligned} \tag{4.14}$$

we can alternatively view the masked condition strategy as implicitly learning an image inpainting model $\nabla_{\mathbf{x}_t} p(\mathbf{y}_{[\mathrm{M}]}|\mathbf{x}_t, \mathbf{y}_{\mathrm{other}})$, a partial caption model $\nabla_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{y}_{\mathrm{other}})$ and a fully conditioned model $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$. The implicit image inpainting model predicts the masked image features $\mathbf{y}_{[\mathrm{M}]}$ using the non-masked image features $\mathbf{y}_{\mathrm{other}}$ and the caption $\mathbf{x}_t$. It enhances the alignment between the masked image features and the current caption. The partial caption model and the fully conditioned model, however, learn to generate the caption with different amounts of conditional information, which improves the performance of the denoising neural network.

Chapter 5

Experiments

5.1 Dataset and Experimental Settings

Experiments are conducted on CUB-200[44] dataset which contains 8855 training images and 2933 test images. Each image includes 10 text descriptions. The average description length is 14.2.

Our VQ-VAE image encoder is taming_f8_8192_openimages_last.pth which comes from the VQGAN[12] trained on OpenImages[25] dataset with codebook size $K = 2886$. It extracts 32× 32 discrete image representations from preprocessed 256×256 images. Our diffusion decoder contains 18 transformer layers with a hidden size equal to 512. The model includes 122M parameters. As for classifier-free guidance and masked condition strategy, we set the data with partial condition rate $P_{\text{part\_cond}}$ and condition mask rate P_{mask} to 0.1.

We set the maximum length as 30 and diffusion step $T = 100$. Our MC-Diffusion is optimized by AdamW[28] on four RTX3090 GPUs with optimizer parameter β set to $(0.9, 0.96)$ and weight decay $\varepsilon = 4.5e-2$. The minimum learning rate is 1.0e-6 and the learning rate increases to 4.5e-4 after 1000 warmup steps. The batch size is set to 32 per GPU and the training epoch is 200.

5.2 Evaluation Metrics

We use the standard reference-based image captioning evaluation metrics BLEU-4[34], Meteor[4], ROUGE[27], CIDEr[42], SPICE[1] and RefClipScore[20] and a reference-free metric CLIPscore[20]. Reference-based metrics evaluate the similarity between the generated text and ground-truth text, while reference-free metrics compute the similarity between the generated caption and input image.

BLEU and ROUGE measure how well the generated text by comparing the overlapping n-grams with the ground-truth text and Meteor addresses the shortcomings of BLEU by considering the accuracy and recall over the entire corpus. CIDEr uses Term Frequency-Inverse Document Frequency (TF-IDF) to assess the importance of each token in the text and also measures the overlap of n-grams. Unlike above-mentioned metrics, SPICE assesses the underlying connection of meaning and semantics of the generated caption and ground-truth caption.

RefClipScore and CLIPscore leverage the CLIP ViT-B/32[35] model to extract CLIP embedding of the generated caption **c**, ground-truth caption **g**, and the input image **i**. RefClipScore and CLIPscore is defined as RefClipS $= 2.5 \times \max(\cos(\mathbf{c},\mathbf{g}),0)$, CLIPscore $= 2.5 \times \max(\cos(\mathbf{c},\mathbf{i}),0)$, respectively.

5.3 Results

5.3.1 Performance Comparison with autoregressive models

Table 5.1 shows the performance comparison. Source Pre-trained and DCC are two autoregressive baselines with a CNN image encoder and LSTM text decoder. ATCIC is the state-of-the-art model on CUB-200 dataset. Both the three approaches are pre-trained on the MSCOCO dataset while our model only pre-trained the VQ-VAE image encoder on the Openimages dataset. Our MC-Diffusion exhibits better performance than autoregressive baselines across all metrics. Specifically, the Meteor and SPICE score of MC-Diffusion is 103.0% and 12.2%, demonstrating an improvement of 1.8% and 1.1% compared to autoregressive baselines. Nevertheless, our MC-Diffusion still can't beat the state-of-the-art model ATCIC with an autoregressive structure on the CUB-200 dataset.

Table5.2 further illustrates the ground truth captions (GT1, GT2, GT3) and generated caption for five images in the test set of the CUB-200 dataset. MC-Diffusion is able to predict relevant captions. Furthermore, it produces more descriptive and longer captions compared to ground truth captions. For the second image in table 5.2, non of the ground-truth captions point out that the bird has black feathers covering the top but our MC-Diffusion. However, the text generated by MC-Diffusion may contain more grammatical errors (e.g. "in with a all" in the first image caption result of table5.2) and have less fluent sentence structure.

5.3.2 Masked Condition Strategy

Table 5.3 and table 5.4 compare the results on the same MC-Diffusion with different guidance scales γ. As shown in tables, as the guidance scale γ increases, when $\gamma \leq 3$, both classifier-

Table 5.1 Result comparison on CUB-200 dataset.

Method	BLEU-4	Meteor	ROUGE	CIDEr	SPICE	RefClipS	CLIPscore
Source Pre-trained	6.1	12.9	33	3	4.6	-	-
DCC[43]	21.4	23.8	46.4	11.9	11.1	-	-
ATCIC[8]	**32.8**	**27.6**	**58.6**	24.8	**13.2**	-	-
Ours	28.1	25.6	54.1	**103**	12.2	79.2	68.0

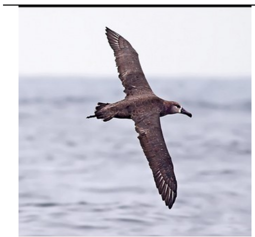
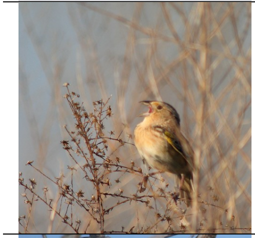
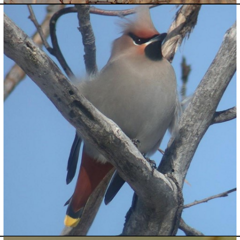
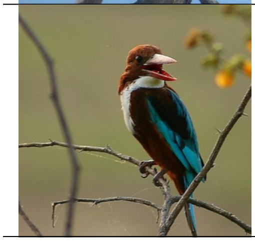
	**MC-Diffusion:** this bird is brown and white in color in with a all and black back and beak. **GT1:** this bird has wings that are black and has a long black bill **GT2:** grey bird with black flat beak with grey and white big wings **GT3:** the dark brown bird has black eye ring and black rectrices.
	**MC-Diffusion:** this small bird has a yellow coloring belly , and a light black specks throughout it ! **GT1:** this bird is brown in color, with a curved beak. **GT2:** this bird has a brown crown, brown primaries, and a brown belly. **GT3:** a small yellow bird with dark spots on its crown and wings
	**MC-Diffusion:** a small bird with red and face has black feathers covering top with . **GT1:** this bird has a grey crown, grey primaries, and a grey belly. **GT2:** this is a grey bird with orange on the crown and cheek patches. **GT3:** the bird has black throat, gray breast, feet, belly and abdomen, it has small beak when compared to its body size.
	**MC-Diffusion:**medium a sized bird with white and brown feathers and a large black beak . **GT1:** this bird has wings that are blue and has a long bill **GT2:** this bird has wings that are black and blue and has a long bill **GT3:** this bird has a brown head a brown body blue wings and a white color around it's neck he also has a very large beak

Table 5.2 Examples of image captioning results generated by MC-Diffusion

free guidance and masked condition strategy exhibit a trade-off between the reference-based metrics and reference-free metric. However, when $\gamma > 3$, the excessively high guidance scale γ may cause the magnitude in the direction from the noise distribution to the real data distribution to be too large, resulting in surpassing the real data distribution and consequently causing both two kinds of evaluation metrics to decline.

To more intuitively compare classifier-free guidance and masked condition strategy, we plot the results from the two tables as Fig5.1. In each subplot, the vertical axis represents the reference-free metric CLIPscore, while the horizontal axis represents various reference-based metrics. Our masked condition strategy performs slightly worse than classifier-free guidance in terms of CLIPscore, with scores of 69.7% and 69.5% when $\gamma = 3$, respectively. However, as the guidance scale increases, during the trade-off process between the two kinds of evaluation metrics, it can be significantly observed that the masked condition strategy shows a much smaller decline in reference-based metrics compared to classifier-free guidance. This may be attributed to our masked condition strategy, which, similar to BERT and MAE, enables the implicit partial condition model $\nabla_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{y}_{\text{other}})$ to learn higher-level language features by masking parts of the information.

Table 5.3 Classifier-free result with different guidance scale γ

Method	BLEU-4	Meteor	ROUGE	CIDEr	SPICE	RefClipS	CLIPscore
$\gamma = 1$	**27**	**25.1**	**53.4**	**103**	**12.4**	**78.9**	67.5
$\gamma = 1.5$	24.7	24.5	51	95	11.7	78.5	68.9
$\gamma = 2$	23	23.2	49.6	86	10.5	77.9	69.5
$\gamma = 3$	21.9	22.5	47.1	83	9.9	77.8	**69.7**
$\gamma = 4$	19.5	22.2	46.3	82	9.7	77.4	68.3

Table 5.4 Masked condition result with different guidance scale γ

Method	BLEU-4	Meteor	ROUGE	CIDEr	SPICE	RefClipS	CLIPscore
$\gamma = 1$	**27.3**	**24.9**	**53.9**	**102**	**12.2**	**79.2**	68.5
$\gamma = 1.5$	25.6	24.1	52.7	96	11.5	78.9	69.3
$\gamma = 2$	24.2	23.2	50.1	93	11.0	78.5	69.4
$\gamma = 3$	23.7	23.0	49.2	91	10.7	78.2	**69.5**
$\gamma = 4$	22.4	22.9	48.9	90	10.6	77.7	69.4

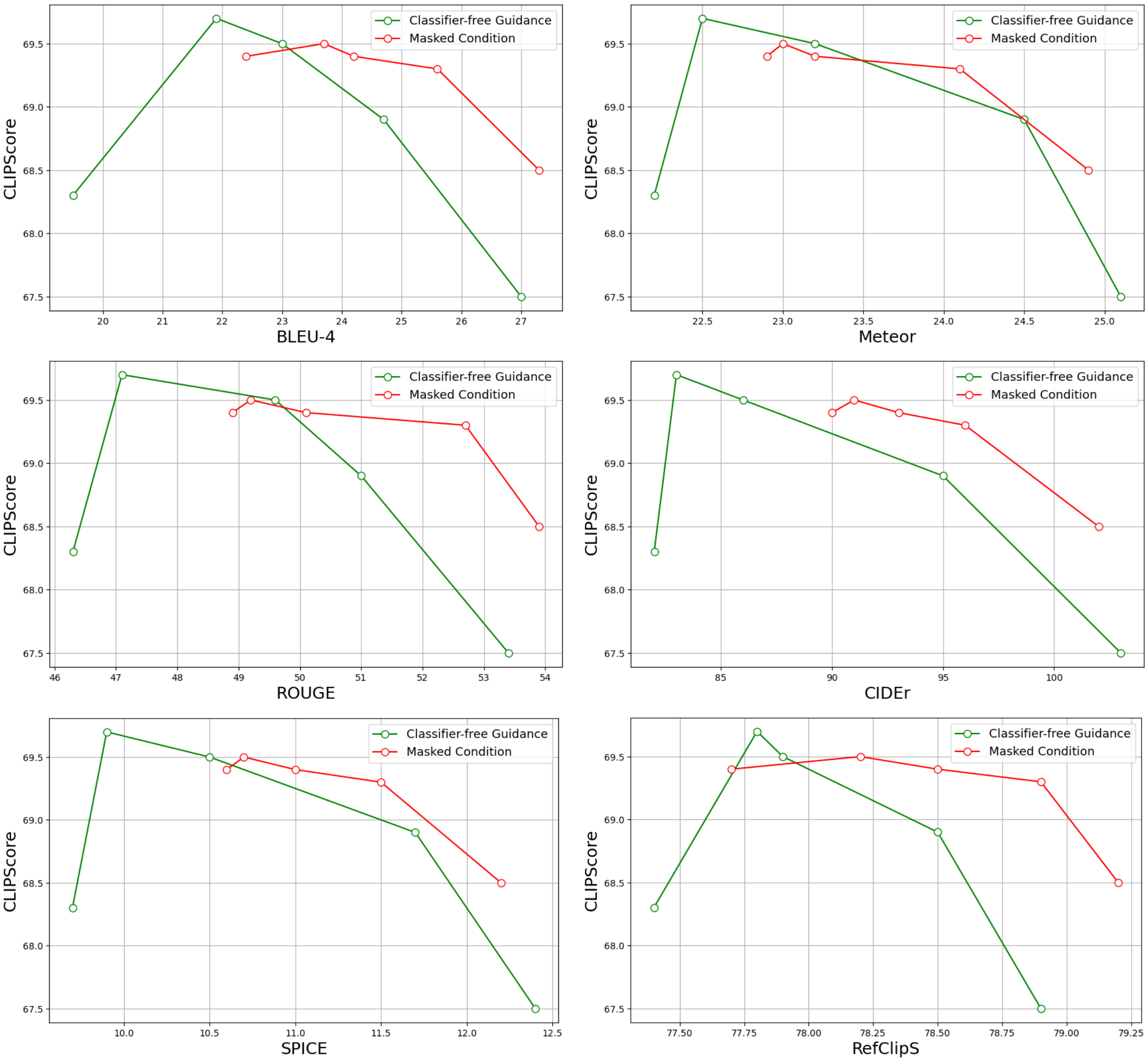


Fig. 5.1 Performance comparison on different guidance scales γ

Chapter 6

Conclusion

In this work, we dive into the idea of encoding the discrete image features in the discrete diffusion model for image captioning. To verify our claim, we devise a masked conditional diffusion model (MC-Diffusion) with a VQ-VAE image encoder and a discrete diffusion decoder. A masked condition strategy is further proposed to better guide the diffusion model. We empirically validate the MC-Diffusion against the autoregressive baselines and state-of-the-art methods on the CUB-200 dataset. At the same time, we also compare the performance of the masked condition strategy and classifier-free guidance on different guidance scales. Although our method is slightly inferior to classifier-free guidance on CLIPscore, we are happy to see that our method alleviates the hurt of reference-based metrics with the increase of guidance scales.

For future work, we can validate our proposed model and methods on more datasets such as MSCOCO. Additionally, since classifier-free guidance in the image generation field also presents a trade-off between image fidelity and diversity, we can test whether the masked condition strategy performs better in this aspect as well.

References

[1] Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2016). Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 382–398. Springer.

[2] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

[3] Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. (2021). Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993.

[4] Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

[5] Barraco, M., Stefanini, M., Cornia, M., Cascianelli, S., Baraldi, L., and Cucchiara, R. (2022). Camel: mean teacher learning for image captioning. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 4087–4094. IEEE.

[6] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[7] Chen, T., ZHANG, R., and Hinton, G. (2022). Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*.

[8] Chen, T.-H., Liao, Y.-H., Chuang, C.-Y., Hsu, W.-T., Fu, J., and Sun, M. (2017). Show, adapt and tell: Adversarial training of cross-domain image captioner. In *Proceedings of the IEEE international conference on computer vision*, pages 521–530.

[9] Cornia, M., Stefanini, M., Baraldi, L., and Cucchiara, R. (2020). Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10578–10587.

[10] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

[11] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634.

[12] Esser, P., Rombach, R., and Ommer, B. (2021). Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883.

[13] Fang, Z., Wang, J., Hu, X., Liang, L., Gan, Z., Wang, L., Yang, Y., and Liu, Z. (2022). Injecting semantic concepts into end-to-end image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18009–18019.

[14] Fei, Z. (2020). Iterative back modification for faster image captioning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3182–3190.

[15] Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., and Rohrbach, M. (2016). Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*.

[16] Gao, J., Meng, X., Wang, S., Li, X., Wang, S., Ma, S., and Gao, W. (2019). Masked non-autoregressive image captioning. *arXiv preprint arXiv:1906.00717*.

[17] Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., and Guo, B. (2022). Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706.

[18] Guo, L., Liu, J., Zhu, X., He, X., Jiang, J., and Lu, H. (2021). Non-autoregressive image captioning with counterfactuals-critical multi-agent learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 767–773.

[19] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

[20] Hessel, J., Holtzman, A., Forbes, M., Bras, R. L., and Choi, Y. (2021). CLIPScore: a reference-free evaluation metric for image captioning. In *EMNLP*.

[21] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

[22] Ho, J. and Salimans, T. (2021). Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

[23] Huang, L., Wang, W., Chen, J., and Wei, X.-Y. (2019). Attention on attention for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4634–4643.

[24] Kornblith, S., Li, L., Wang, Z., and Nguyen, T. (2023). Classifier-free guidance makes image captioning models more descriptive. In *ICLR 2023 Workshop on Multimodal Representation Learning: Perks and Pitfalls*.

[25] Krasin, I., Duerig, T., Alldrin, N., Veit, A., Abu-El-Haija, S., Belongie, S., Cai, D., Feng, Z., Ferrari, V., Gomes, V., Gupta, A., Narayanan, D., Sun, C., Chechik, G., and Murphy, K. (2016). Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*.

[26] Li, J., Li, D., Savarese, S., and Hoi, S. (2023). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

[27] Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

[28] Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[29] Luo, C. (2022). Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*.

[30] Luo, J., Li, Y., Pan, Y., Yao, T., Feng, J., Chao, H., and Mei, T. (2023). Semantic-conditional diffusion networks for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23359–23368.

[31] Luo, Y., Ji, J., Sun, X., Cao, L., Wu, Y., Huang, F., Lin, C.-W., and Ji, R. (2021). Dual-level collaborative transformer for image captioning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 2286–2293.

[32] Mao, J., Xu, W., Yang, Y., Wang, J., and Yuille, A. L. (2014). Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*.

[33] Pan, Y., Yao, T., Li, Y., and Mei, T. (2020). X-linear attention networks for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10971–10980.

[34] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

[35] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

[36] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

[37] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR.

[38] Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

[39] Tang, Z., Gu, S., Bao, J., Chen, D., and Wen, F. (2022). Improved vector quantized diffusion models. *arXiv preprint arXiv:2205.16007.*

[40] Van Den Oord, A., Vinyals, O., et al. (2017). Neural discrete representation learning. *Advances in neural information processing systems*, 30.

[41] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[42] Vedantam, R., Lawrence Zitnick, C., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

[43] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

[44] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. J. (2011). The caltech-ucsd birds-200-2011 dataset.

[45] Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C., and Wang, L. (2022). Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100.*

[46] Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., and Cao, Y. (2021). Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904.*

[47] Yan, X., Fei, Z., Li, Z., Wang, S., Huang, Q., and Tian, Q. (2021). Semi-autoregressive image captioning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2708–2716.

[48] Yu, H., Liu, Y., Qi, B., Hu, Z., and Liu, H. (2023). End-to-end non-autoregressive image captioning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

[49] Zhang, X., Sun, X., Luo, Y., Ji, J., Zhou, Y., Wu, Y., Huang, F., and Ji, R. (2021). Rstnet: Captioning with adaptive attention on visual and non-visual words. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15465–15474.

[50] Zhu, Z., Wei, Y., Wang, J., Gan, Z., Zhang, Z., Wang, L., Hua, G., Wang, L., Liu, Z., and Hu, H. (2022). Exploring discrete diffusion models for image captioning. *arXiv preprint arXiv:2211.11694.*